

Refine Search

Search Results -

Terms	Documents
L9 and (row with ((key or comparand) near4 (width or length)))	7

Database:

- US Pre-Grant Publication Full-Text Database
- US Patents Full-Text Database
- US OCR Full-Text Database
- EPO Abstracts Database
- JPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search:

Search History

DATE: Wednesday, August 17, 2005 [Printable Copy](#) [Create Case](#)

<u>Set</u>	<u>Name</u>	<u>Query</u>	<u>Hit</u>	<u>Set</u>
			<u>Count</u>	<u>Name</u>
	side by side			result set
		DB=USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
<u>L10</u>		L9 and (row with ((key or comparand) near4 (width or length)))	7	<u>L10</u>
<u>L9</u>		CAM and ((row or cell or block) near4 (segment\$4 or partition\$4 or section\$4 or split\$4))	7528	<u>L9</u>
		DB=PGPB,USPT; PLUR=YES; OP=OR		
<u>L8</u>		L7 and (row with ((key or comparand) near4 (width or length)))	2	<u>L8</u>
<u>L7</u>		L4 and ((row or cell or block) near4 (segment\$4 or partition\$4 or section\$4 or split\$4))	104	<u>L7</u>
<u>L6</u>		L4 and ((column of row) same (row near4 (segment\$4 or partition\$4 or section\$4 or split\$4)))	0	<u>L6</u>
<u>L5</u>		L4 and column	185	<u>L5</u>
<u>L4</u>		L3 and switch\$3	212	<u>L4</u>
<u>L3</u>		L2 and (CAM near4 (block or cell))	369	<u>L3</u>
<u>L2</u>		compar\$6 near6 row and CAM	995	<u>L2</u>
<u>L1</u>		((20020126672 or 5619713 or 20030131331 or 6169685 or 6374326 or 5890005).pn.) and (row or column or matrix))	3	<u>L1</u>

END OF SEARCH HISTORY

Search Results[BROWSE](#)[SEARCH](#)[IEEE XPLOR GUIDE](#)

Results for "((cam and (row <near/4> (segment\$ or partition\$ or section\$ or split\$) <sentence> ((widt...".

[E-mail](#)

Your search matched 0 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.[» Search Options](#)[View Session History](#)[Modify Search](#)[New Search](#) [»](#) Check to search only within this results setDisplay Format: Citation Citation & Abstract[» Key](#)

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistance.

[Help](#) [Contact Us](#) [Privacy &](#)

© Copyright 2005 IEEE -

Terms used

content addressable memory or CAM and row near/4 segment or partition or section or split sentence width

 Sort results by relevance 
 [Save results to a Binder](#)
[Try an Advanced Search](#)

 Display results expanded form 
 [Search Tips](#)
[Try this search](#)
 [Open results in a new window](#)

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

1 Pen computing: a technology overview and a vision

André Meyer

 July 1995 **ACM SIGCHI Bulletin**, Volume 27 Issue 3

 Full text available:  [pdf\(5.14 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This work gives an overview of a new technology that is attracting growing interest in public as well as in the scientific community. The visible difference from other technologies is in the use of a pen or pencil as the primary means of input. The pen is used as a pointing device, as a drawing tool, as a means of interacting with a computer, picking up the familiar pen and paper interface metaphor. From this follows a set of conclusions and recommendations for the future development of pen computing. Starting with a short historic ...

2 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

 November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Computer Communication**

 Full text available:  [pdf\(4.21 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process flow diagrams are often not sufficient to gain a better understanding of the execution of the application. The visualization tool we use is based on the University of Waterloo. However, these diagrams are often very complex and do not provide the user with a clear overview of the application. In our experience, such tools display repeated occurrences of non-trivial communication patterns, which are often the cause of performance problems. In this paper, we propose a new visualization technique that ...

3 Scalable high-speed prefix matching

Marcel Waldvogel, George Varghese, Jon Turner, Bernhard Plattner

 November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

 Full text available:  [pdf\(933.02 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Finding the longest matching prefix from a database of keywords is an old problem with a number of different solutions, from simple dictionary searches to advanced memory management to computational geometry. But perhaps the most interesting aspect of this problem is that it is becoming increasingly important in the Internet, when forwarding packets from router to router. Internet traffic is increasing; at the same time, a growing user population is increasing the size of routing tables and the number of prefixes to be matched. In this paper, we propose a new algorithm for prefix matching that is both ...

Keywords: collision resolution, forwarding lookups, high-speed networking

4 Face recognition: A literature survey

W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld

 December 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 4

 Full text available:  [pdf\(4.28 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

As one of the most successful applications of image analysis and understanding, face recognition has been the focus of much research and development over the past several years. At least two reasons account for this trend: the first is the increasing number of applications, particularly in the commercial and law enforcement applications, and the second is the availability of feasible techniques. Even though current machine recognition systems have reached a certain level of maturity, their performance is still far from perfect. In this survey, we ...

5 Special issue: AI in engineering

D. Sriram, R. Joobhani

January 1985 **ACM SIGART Bulletin**, Issue 91

Full text available:  [pdf\(8.79 MB\)](#)

Additional Information: [full citation](#), [abstract](#)

The papers in this special issue were compiled from responses to the announcement in the July 1984 issue of *ACM SIGART Bulletin* and notices posted over the ARPAnet. The interest being shown in this area is reflected in the sixteen countries. About half the papers were received over the computer network.

6 A pipelined memory architecture for high throughput network processors

Timothy Sherwood, George Varghese, Brad Calder

May 2003 **ACM SIGARCH Computer Architecture News, Proceedings of the 30th annual Computer architecture**, Volume 31 Issue 2

Full text available:  [pdf\(213.66 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Designing ASICs for each new generation of backbone routers is a time intensive and fiscally drain-ing task. Instead, we propose a pipelined memory architecture for backbone routers, based on the manipulation of memory blocks. This can provide a feasible design alternative to custom ASICs. We propose a pipelined memory design that trades off memory bandwidth and latency for high throughput. We co-explore architectural tradeoffs with the design of several important components.

7 Interactive Editing Systems: Part II

Norman Meyrowitz, Andries van Dam

September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3

Full text available:  [pdf\(9.17 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

8 Parallel execution of prolog programs: a survey

Gopal Gupta, Enrico Pontelli, Khayri A.M. Ali, Mats Carlsson, Manuel V. Hermenegildo

July 2001 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 23 Issue 3

Full text available:  [pdf\(1.95 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Since the early days of logic programming, researchers in the field realized the potential for exploiting parallelism in logic programs. Their high-level nature, the presence of nondeterminism, and their declarative characteristics, make logic programs interesting candidates for obtaining speedups through parallel execution. However, the characteristics of logic programs also pose significant challenges for parallel execution. In fact, the typical applications of logic programming frequently involve irregular computation patterns, such as backtracking and non-determinism, which are difficult to parallelize. In this survey, we review the state of the art in parallel execution of logic programs, with a focus on parallel execution of Prolog programs. We discuss the main parallel execution paradigms for logic programs, including parallel Prolog, parallel Prolog with shared memory, and parallel Prolog with distributed memory. We also discuss the main challenges and open problems in parallel execution of logic programs, and the future directions for research in this area.

Keywords: Automatic parallelization, constraint programming, logic programming, parallelism, parallel execution, parallel Prolog

9 Memory-efficient state lookups with fast updates

Sandeep Sikka, George Varghese

August 2000 **ACM SIGCOMM Computer Communication Review, Proceedings of the conference on Technologies, Architectures, and Protocols for Computer Communication**, Volume 30 Issue 3

Full text available:  [pdf\(384.82 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Routers must do a best matching prefix lookup for every packet; solutions for Gigabit speeds are not feasible. In this paper, we seek a scalable solution whose speed scales with memory speeds while allowing large numbers of lookups. We show that providing such a solution requires careful attention to memory allocation and pipelining. We propose a memory-efficient state lookup mechanism that uses on-chip or off-chip SRAM which is limited by either expense or size.

10 Design and Implementation of High-Performance Memory Systems for Future Packet Buffer

Jorge García, Jesús Corbal, Llorenç Cerdà, Mateo Valero

December 2003 **Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture**

Full text available:  [pdf\(348.55 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

In this paper we address the design of a future high-speed router that supports line rates as high as 100 Gbps and 100 ports. The router must support several service classes. Building such a high-speed router would raise many technical challenges, being the packet buffer design, mainly because in router design it is important to provide worst-case performance guarantees. A previous packet buffer design provides worst-case bandwidth guarantees, but it is not suitable for a high-speed router. In this paper, we propose a new packet buffer design that provides worst-case bandwidth guarantees and is suitable for a high-speed router. The proposed design is based on a combination of a memory-based buffer and a memory-based buffer. The proposed design is able to provide worst-case bandwidth guarantees and is suitable for a high-speed router.

11 Distributed operating systems

Andrew S. Tanenbaum, Robbert Van Renesse

December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4

Full text available:  [pdf\(5.49 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Distributed operating systems have many aspects in common with centralized ones, but they also intended as an introduction to distributed operating systems, and especially to current university discussion of what constitutes a distributed operating system and how it is distinguished from a c issues are discussed. Then several examples of current research projects are examined in some d

12 A PDP-8 emulator program

Brian J. Shelburne

March 2002 **Journal on Educational Resources in Computing (JERIC)**, Volume 2 Issue 1

Full text available:  [pdf\(270.03 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

The clean, simple, and elegant architecture of the classic PDP-8 makes it an ideal candidate for st organization. The PDP-8 emulator program allows a user to write, edit, assemble, debug, trace, a PDP-8 assembler language programs. With it, the user can obtain a feel for the PDP-8. The PDP-8 built-in text editor which is used to write and edit PDP-8 assembler language programs, an assem

Keywords: Computer architecture simulator, education

13 The Vector-Thread Architecture

Ronny Krashinsky, Christopher Batten, Mark Hampton, Steve Gerding, Brian Pharris, Jared Casper, K March 2004 **ACM SIGARCH Computer Architecture News , Proceedings of the 31st annual Computer architecture ISCA '04**, Volume 32 Issue 2

Full text available:  [pdf\(317.13 KB\)](#)

Additional Information: [full citation](#), [abstract](#)

The vector-thread (VT) architectural paradigm unifies the vector and multithreaded compute mode programmer with a control processor and a vector of virtual processors (VPs). The control process broadcast instructions to all the VPs or each VP can use thread-fetches to direct its own control flow. The vector and threaded control mechanisms allows a VT architecture to flexibly and compactly encode

14 Applications II: Towards automatic analysis of social interaction patterns in a nursing home

Datong Chen, Jie Yang, Howard D. Wactlar

October 2004 **Proceedings of the 6th ACM SIGMM international workshop on Multimedia in**

Full text available:  [pdf\(490.67 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index](#)

In this paper, we propose an ontology-based approach for analyzing social interaction patterns in a nursing home. The interaction patterns are broken into individual activities and behavior events using a multi-level classification. To take advantage of an ontology in representing how social interactions evolve, we design and realize an ontology for social interactions. We collected and analyzed 100 hours of video recorded in the public spaces of a nursing home. The ontology captures the knowledge gained from 80 hours of video recorded in the public spaces of a nursing home. The ontology is used to analyze the social interaction patterns in a nursing home.

Keywords: human activity, medical care, ontology, social interaction, stochastic modeling

15 On-line Text Editing: A Survey

Andries van Dam, David E. Rice

September 1971 **ACM Computing Surveys (CSUR)**, Volume 3 Issue 3

Full text available:  [pdf\(1.91 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

This paper is a survey of current methods for the on-line creation and editing of computer programs. The characteristics of on-line editing systems are examined and examples of various implementation categories: program editors, text editors, and terminals with local editing facilities.

16 String storage and searching for data base applications: Implementation on the INDY back-end

George P. Copeland

August 1978 **Proceedings of the fourth workshop on Computer architecture for non-numerical applications**

Full text available:  [pdf\(854.23 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

User and hardware cost trends dictate that data base systems should provide more complete func

reliability by increasing the amount of hardware present in the system. These goals are accomplished within a one-dimensional cellular storage system called INDY. The INDY backend is intended to provide functionality for implementing all data models. The INDY cellular storage array is intended to provide functionality

17 Translator writing systems

Jerome Feldman, David Gries

February 1968 **Communications of the ACM**, Volume 11 Issue 2

Full text available:  pdf(4.47 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

A critical review of recent efforts to automate the writing of translators of programming language syntax and its application to translator writing are discussed in Section II. Various approaches to (semantic) aspects of translator writing are discussed in Section III, and several related topics in

Keywords: compiler compiler-compiler, generator, macroprocessor, meta-assembler, metacompiler, syntax, syntax-directed, translator, translator writing system

18 String storage and searching for data base applications: implementation on the INDY backend

George P. Copeland

August 1978 **ACM SIGMOD Record , ACM SIGIR Forum , ACM SIGARCH Computer Architect**

2

Full text available:  pdf(986.51 KB)

Additional Information: [full citation](#), [abstract](#), [references](#)

User and hardware cost trends dictate that data base systems should provide more complete functionality by increasing the amount of hardware present in the system. These goals are accomplished within a one-dimensional cellular storage system called INDY. The INDY backend is intended to provide functionality for implementing all data models. The INDY cellular storage array is intended to provide functionality

19 Parallel algorithms for data compression

M. E. Gonzalez Smith, J. A. Storer

April 1985 **Journal of the ACM (JACM)**, Volume 32 Issue 2

Full text available:  pdf(1.99 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Parallel algorithms for data compression by textual substitution that are suitable for VLSI implementation. "dynamic" dictionary schemes are considered.

20 Conference abstracts

January 1977 **Proceedings of the 5th annual ACM computer science conference**

Full text available:  pdf(3.14 MB)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

One problem in computer program testing arises when errors are found and corrected after a program is run. How can it be shown that a fix to one area of the code does not adversely affect the execution of the program? A quantitative method for assuring that new program modifications do not introduce new errors into the program is presented. The test philosophy that every program instruction that could possibly be reached and tested from the start of the program is used.

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2003 ACM, Inc. All rights reserved.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [RealPlayer](#)